UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/994,574	11/26/2001	William R. Wheeler	10559/602001/P12886	7301

20985 7590 11/01/2005

FISH & RICHARDSON, PC
12390 EL CAMINO REAL
SAN DIEGO, CA 92130-2081

EXAMINER	
GUILL, RUSSELL L	
ART UNIT	**PAPER NUMBER**
2123	

DATE MAILED: 11/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/994,574	WHEELER ET AL.	
	Examiner	Art Unit	
	Russell L. Guill	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 August 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☐ Claim(s) _____ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-7,9-18 and 20-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *9/17/2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to an Amendment filed August 8, 2005. Claims 2, 8 and 19 have been canceled. Claims 1, 3, 7, 9, 15, 18, 20 and 25 have been amended. Claims 1, 3 - 7, 9 - 18 and 20 - 25 have been examined. Claims 1, 3 - 7, 9 - 18 and 20 - 25 have been rejected.

Response to Arguments

2. Applicant's arguments with respect to all claims have been considered but are moot in view of the new ground(s) of rejection necessitated by amendments.
3. Regarding Applicant's section 35 USC § 101 in the REMARKS, Applicant's amendments to the claims overcome the rejections under 35 USC § 101, and the rejections are withdrawn.
4. It appears that the Applicant intended to amend claim 15 to overcome a rejection under 35 U.S.C. 112, second paragraph, but the amendment was not applied to the claim (Applicant's Remarks, page 9, section 35 USC § 112). The intended amendment was to swap the phrase "central databases" with "central database." Accordingly, the rejection is maintained.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5.1. Claim 7 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claim recites, "automatically updating the logic design with the updated value of the defined signal parameter when the value of the defined signals is updated in the central database". The phrase, "the defined signals" lacks antecedent basis. For the purpose of claim examination, the phrase, "the defined signals" is interpreted as "the defined signal parameter". Correction or amendment is required.

5.2. Claim 15 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claim recites, "the central databases". There is insufficient antecedent basis for the term. For the purpose of claim interpretation, the phrase, "the central databases" is interpreted as "the central database". Correction or amendment is required.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

8. Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) in view of Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

8.1. The art of ExpressiveSystems is directed to logic design systems ***(page 2)***.

8.2. The art of Yamagishi is directed to logic design systems ***(page 13.2.1, Abstract, and section 1 Introduction)***.

8.3. ExpressiveSystems appears to teach:

8.3.1. a logic design module operable to be used by one or more users to generate a logic design as part of an electrical circuit ***(page 2, first and second paragraphs)***.

8.3.2. modifiable signal parameters that are accessible for use by the users of the logic design module ***(page 8 and page 9)***.

8.3.3. The logic design module is operable to automatically update the logic design when the signal parameters are modified (**page 4, text and graphic inset figure located in the lower-right portion of the page; pages 8 – 9, especially the fourth paragraph that starts with "Clicking on signal names . . .", second sentence: "Double clicking opens the signal editor allowing detailed changes to be made and *applied to the local level or the entire design*"; page 10, Code Generation; pages 21 – 25, please note the automatic code generation after modifying signal parameters, and that adding new signal parameters constitutes modification of signal parameters in the database of signal parameters**).

8.4. ExpressiveSystems does not specifically teach:

8.4.1. ***a central database integrated with the logic design module*** and including modifiable signal parameters that are accessible to use by the users of the logic design module in the logic design tasks.

8.4.2. The logic design module is operable to automatically update the logic design when the signal parameters ***in the central database*** are modified

8.5. Yamagishi appears to teach:

8.5.1. a central database integrated with the logic design module ***(pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)***.

8.6. ExpressiveSystems and Yamagishi are analogous art because they are both directed to the art of logic design systems.

8.7. The motivation to use the art of Yamagishi with the art of ExpressiveSystems would have been obvious given the statement in Yamagishi that the database allows much closer integration among tools than other commercially available frameworks ***(page 13.2.3, left-side of page, lines 1 – 3)***, and the expressed benefit in ExpressiveSystems of the software to save designers time and effort by simplifying the maintenance of the design ***(page 2, second paragraph)***.

9. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference) in view of IEEEVerilog (IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language, 1995, IEEE).

9.1. Claim 3 is a dependent claim of claim 1, and thereby inherits all of the rejected limitations of claim 1.

9.2. The art of ExpressiveSystems is directed to logic design systems ***(page 2)***.

9.3. The art of IEEEVerilog is directed to logic design ***(page iii, section Introduction)***.

9.4. ExpressiveSystems does not specifically teach that the logic design module is structured and arranged to indicate design discrepancies automatically in the logic design resulting from the modifications to the signal parameters in the central database.

9.5. Yamagishi appears to teach a central database integrated with the logic design module ***(pages 13.2.2 and 13.2.3, section 2.2 Database FALNET).***

9.6. IEEEVerilog appears to teach indicating design discrepancies automatically in the logic design resulting from the modifications to the signal parameters ***(page 59, lines 1 – 25)***.

9.6.1. Regarding ***(page 59, lines 1 – 25)***; it is obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

9.7. ExpressiveSystems and IEEEVerilog are analogous art because they both include the problem of logic design.

9.8. The motivation to use the art of IEEEVerilog with the art of ExpressiveSystems would have been obvious since:

9.8.1. ExpressiveSystems generates Verilog computer code ***(page 24 and page 25)***, and IEEEVerilog validates and executes Verilog computer code, and

9.8.2. given the expressed benefit in ExpressiveSystems that the software saves designers time and effort by simplifying the maintenance of the design ***(page 2, second paragraph)***, and

9.8.3. indicating design discrepancies earlier in the design before generating Verilog computer code would save time and effort.

9.9. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of ExpressiveSystems with the art of IEEEVerilog to produce the claimed invention.

10. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference) and IEEEVerilog (IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language, 1995, IEEE).

10.1. Claim 4 is a dependent claim of claim 3, and thereby inherits all of the rejected limitations of claim 3.

10.2. ExpressiveSystems appears to teach indicating a bit width ***(page 24, screen displayed at top of page, the bit width is embedded in the line)***.

10.3. ExpressiveSystems does not specifically teach indicating a bit width ***error***.

10.4. IEEEVerilog appears to teach indicating a bit width error ***(page 59, lines 1 – 25)***.

10.4.1. Regarding ***(page 59, lines 1 – 25)***; it is obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

10.5. Therefore, as discussed above, it would have been obvious to use the art of IEEEVerilog with the art of ExpressiveSystems to obtain the invention of claim **4**.

11. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

11.1. Claim 5 is a dependent claim of claim 1, and thereby inherits all of the rejected limitations of claim 1.

11.2. ExpressiveSystems appears to teach that the signal parameters include a signal bit width and a value for the signal bit width ***(page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32. This may be confirmed by examining the code generated on page 25)***.

12. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

12.1. Claim 6 is a dependent claim of claim 1, and thereby inherits all of the rejected limitations of claim 1.

12.2. ExpressiveSystems appears to teach that the signal parameters include a signal bit position and a value for the signal bit position ***(page 14, section labeled 'Flexible signal handling')***.

13. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) in view of

Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

13.1. The art of ExpressiveSystems is directed to logic design systems ***(page 2)***.

13.2. The art of Yamagishi is directed to logic design systems ***(page 13.2.1, Abstract, and section 1 Introduction)***.

13.3. ExpressiveSystems appears to teach:

13.3.1. Defining a signal parameter with a value ***(page 8 and page 9)***.

13.3.2. Maintaining the defined signal value ***(page 8 and page 9)***.

13.3.3. Using the defined signal parameter in computer code for a logic design forming part of an electrical circuit ***(page 23 and page 24 and page 25 - page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32)***.

13.3.3.1. Regarding ***(page 23 and page 24 and page 25 - page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32)***; the pages recited demonstrate defining a signal parameter followed by generation of computer code for logic design.

13.3.4. Updating the value of a defined signal parameter (**pages 8 – 9**).

13.3.5. Automatically updating a logic design with the updated value of a defined signal parameter when the value of the defined signals is updated (**page 4, text and graphic inset figure located in the lower-right portion of the page; pages 8 – 9, especially the fourth paragraph that starts with "Clicking on signal names . . .", second sentence: "Double clicking opens the signal editor allowing detailed changes to be made and *applied to the local level or the entire design*"; page 10, Code Generation; pages 21 – 25, please note the automatic code generation after modifying signal parameters, and that adding new signal parameters constitutes modification of signal parameters in the database of signal**

parameters. It would have been obvious that the computer code for logic design is generated using the stored value of the signal parameter, so that the computer code for logic design would be generated using an updated signal parameter).

13.4. ExpressiveSystems does not specifically teach:

13.4.1. Maintaining the defined signal value ***in a central database***.

13.4.2. Using the defined signal parameter ***that is maintained in the central database*** in computer code for a logic design forming part of an electrical circuit.

13.4.3. Updating the value of a defined signal parameter ***in the central database***.

13.4.4. Automatically updating a logic design with the updated value of a defined signal parameter when the value of the defined signals is updated ***in the central database***.

13.5. Yamagishi appears to teach a central database integrated with the logic design module ***(pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)***.

13.6. ExpressiveSystems and Yamagishi are analogous art because they are both directed to the art of logic design systems.

13.7. The motivation to use the art of Yamagishi with the art of ExpressiveSystems would have been obvious given the statement in Yamagishi that the database allows much closer integration among tools than other commercially available frameworks ***(page 13.2.3, left-side of page, lines 1 – 3)***, and the expressed benefit in ExpressiveSystems of the software to save designers time and effort by simplifying the maintenance of the design ***(page 2, second paragraph)***.

13.8. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Yamagishi with the art of ExpresiveSystems to obtain the claimed invention.

14. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference) in view of IEEEVerilog (IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language, 1995, IEEE).

14.1. Claim 9 is a dependent claim of claim 7, and thereby inherits all of the rejected limitations of claim 7.

14.2. The art of ExpressiveSystems is directed to logic design systems ***(page 2)***.

14.3. The art of IEEEVerilog is directed to logic design ***(page iii, section Introduction)***.

14.4. ExpressiveSystems does not specifically teach automatically indicating design discrepancies in the logic design that result from updating the value of the defined signal parameter.

14.5. IEEEVerilog appears to teach indicating design discrepancies automatically in the logic design resulting from the modifications to the signal parameters ***(page 59, lines 1 – 25)***.

14.5.1. Regarding ***(page 59, lines 1 – 25)***; it is obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

14.6. ExpressiveSystems and IEEEVerilog are analogous art because they both are directed to the problem of logic design.

14.7. The motivation to use the art of IEEEVerilog with the art of ExpressiveSystems would have been obvious since:

14.7.1. ExpressiveSystems generates Verilog computer code ***(page 24 and page 25)***, and IEEEVerilog executes Verilog computer code, and

14.7.2. given the expressed benefit in ExpressiveSystems that the software saves designers time and effort by simplifying the maintenance of the design ***(page 2, second paragraph)***, and

14.7.3. indicating design discrepancies earlier in the design before generating Verilog computer code would save time and effort.

14.8. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of ExpressiveSystems with the art of IEEEVerilog to produce the claimed invention.

15. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference) in view of IEEEVerilog (IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language, 1995, IEEE).

15.1. Claim 10 is a dependent claim of claim 9, and thereby inherits all of the rejected limitations of claim 9.

15.2. ExpressiveSystems appears to teach graphically indicating a bit width ***(page 24, screen displayed at top of page, the bit width is embedded in the line)***.

15.3. ExpressiveSystems does not specifically teach indicating a bit width ***error***.

15.4. IEEEVerilog appears to teach indicating a bit width error ***(page 59, lines 1 – 25)***.

15.4.1. Regarding ***(page 59, lines 1 – 25)***; it is obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

15.5. Therefore, as discussed above, it would have been obvious to use the art of IEEEVerilog with the art of ExpressiveSystems to obtain the invention of claim 10.

16. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

16.1. Claim 11 is a dependent claim of claim 7, and thereby inherits all of the rejected limitations of claim 7.

16.2. ExpressiveSystems appears to teach that the signal parameter includes a signal bit width and the value includes a value for the signal bit width ***(page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32. This may be confirmed by examining the code generated on page 25)***.

17. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

17.1. Claim 12 is a dependent claim of claim 7, and thereby inherits all of the rejected limitations of claim 7.

17.2. ExpressiveSystems appears to teach that the signal parameter includes a signal bit position and the value includes a value for the signal bit position ***(page 14, section labeled 'Flexible signal handling', 'Multi-bit signals can be created with non-zero starting indexes', and page 8, on the signal display window, in the column labeled 'Signal Name', the signal named IO_DATA[15:0])***.

18. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

18.1. Claim 13 is a dependent claim of claim 7, and thereby inherits all of the rejected limitations of claim 7.

18.2. ExpressiveSystems appears to teach that the signal parameter includes a bit field and the value includes a value for the bit field ***(page 14, section labeled 'Flexible signal handling', 'Multi-bit signals can be created with non-zero starting indexes', and page 8, on the signal display window, in the column labeled 'Signal Name', the signal named IO_DATA[15:0])***.

19. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

19.1. Claim 14 is a dependent claim of claim 7, and thereby inherits all of the rejected limitations of claim 7.

19.2. Yamagishi appears to teach accessing the central database by one or more users ***(page 13.2.1, figure 1)***.

20. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) in view of Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

20.1. The art of ExpressiveSystems is directed to logic design systems ***(page 2)***.

20.2. The art of Yamagishi is directed to logic design systems ***(page 13.2.1, Abstract, and section 1 Introduction)***.

20.3. ExpressiveSystems appears to teach:

20.3.1. Defining a signal parameter with a value ***(page 8 and page 9)***.

20.3.2. Defining one or more signal parameters ***(page 8 and page 9)***.

20.3.3. A value for the signal parameters ***(page 8 and page 9)***.

20.3.4. A logic design module that uses the signal parameters in a logic design forming part of an electrical circuit ***(page 23 and page 24 and page 25 - page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32)***.

20.3.4.1. Regarding ***(page 23 and page 24 and page 25 - page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32)***; the pages recited demonstrate defining a signal parameter followed by generation of computer code for logic design.

20.3.5. A value for the defined signal parameters is operable to be modified (***pages 8 – 9***).

20.3.6. The logic design is automatically updated with the modified value of the defined signal parameters when the value for the defined signal parameters is modified (**page 4, text and graphic inset figure located in the lower-right portion of the page; pages 8 – 9, especially the fourth paragraph that starts with "Clicking on signal names . . .", second sentence: "Double clicking opens the signal editor allowing detailed changes to be made and *applied to the local level or the entire design*"; page 10, Code Generation; pages 21 – 25, please note the automatic code generation after modifying signal parameters, and that adding new signal parameters constitutes modification of signal parameters in the database of signal parameters**).

20.4. ExpressiveSystems does not specifically teach:

20.4.1. ***A central database accessible by one or more users.***

20.4.2. One or more signal parameters defined ***in the central database***.

20.4.3. ***an interface between the central databases and*** a logic design module that uses the signal parameters in a logic design.

20.4.4. A value for the defined signal parameters ***in the central database*** is operable to be modified.

20.4.5. The logic design is automatically updated with the modified value of the defined signal parameters when the value for the defined signal parameters ***in the central database*** is modified.

20.5. Yamagishi appears to teach a central database accessible by one or more users ***(page 13.2.1, figure 1, and pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)***.

20.6. Yamagishi appears to teach logic design data stored in a central database ***(page 13.2.1, section 2 FALcyber)***.

20.7. Yamagishi appears to teach an interface between the central databases and a logic design module ***(pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)***.

20.8. ExpressiveSystems and Yamagishi are analogous art because they are both directed to the art of logic design systems.

20.9. The motivation to use the art of Yamagishi with the art of ExpressiveSystems would have been obvious given the statement in Yamagishi that the database allows much closer integration among tools than other commercially available frameworks ***(page 13.2.3, left-side of page, lines 1 – 3)***, and the expressed benefit in ExpressiveSystems of the software to save designers time and effort by simplifying the maintenance of the design ***(page 2, second paragraph)***.

20.10. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Yamagishi with the art of ExpresiveSystems to obtain the claimed invention.

21. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

21.1. Claim 16 is a dependent claim of claim 15, and thereby inherits all of the rejected limitations of claim 15.

21.2. ExpressiveSystems appears to teach that the signal parameters include a signal bit width and the value includes a value for the signal bit width ***(page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32. This may be confirmed by examining the code generated on page 25)***.

22. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

22.1. Claim 17 is a dependent claim of claim 15, and thereby inherits all of the rejected limitations of claim 15.

22.2. ExpressiveSystems appears to teach that the signal parameters include a signal bit position and the value includes a value for the signal bit position ***(page 14, section labeled 'Flexible signal handling', 'Multi-bit signals can be created with non-zero starting indexes', and page 8, on the signal display window, in the column labeled 'Signal Name', the signal named IO_DATA[15:0])***.

23. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) in view of Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

23.1. The art of ExpressiveSystems is directed to logic design systems ***(page 2)***.

23.2. The art of Yamagishi is directed to logic design systems ***(page 13.2.1, Abstract, and section 1 Introduction)***.

23.3. ExpressiveSystems appears to teach:

23.3.1. Defining a signal parameter with a value ***(page 8 and page 9)***.

23.3.2. Maintaining the defined signal parameter ***(page 8 and page 9)***.

23.3.3. Using the defined signal parameter in computer code for a logic design forming part of an electrical circuit ***(page 23 and page 24 and page 25 - page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32)***.

23.3.3.1. Regarding ***(page 23 and page 24 and page 25 - page 23, on the Net Editor screen, in the field labeled New Signal(s), a signal is named dBus with a width of 32)***; the pages recited demonstrate defining a signal parameter of the signal database followed by generation of computer code for logic design.

23.3.4. Updating the value of a defined signal parameter (**pages 8 – 9**).

23.3.5. Automatically updating a logic design with the updated value of a defined signal parameter when the value of the defined signal parameter is updated (**page 4, text and graphic inset figure located in the lower-right portion of the page; pages 8 – 9, especially the fourth paragraph that starts with "Clicking on signal names . . .", second sentence: "Double clicking opens the signal editor allowing detailed changes to be made and *applied to the local level or the entire design*"; page 10, Code Generation; pages 21 – 25, please note the automatic code generation after modifying signal parameters, and that adding new signal parameters constitutes modification of signal parameters in the database of signal parameters. It would have been obvious that the computer code for logic design is generated using the stored value of the signal parameter, so that the computer code for logic design would be generated using an updated signal parameter**).

23.4. ExpressiveSystems does not teach specifically teach:

23.4.1. Using the defined signal parameter ***that is maintained in the central database*** in computer code for a logic design forming part of an electrical circuit.

23.4.2. Maintaining the defined signal parameter ***in a central database***.

23.4.3. Updating the value of a defined signal parameter ***in the central database***.

23.4.4. Automatically updating a logic design with the updated value of a defined signal parameter when the value of the defined signal parameter is updated ***in the central database***.

23.5. Yamagishi appears to teach maintaining logic design data in a central database ***(page 13.2.1, section 2 FALcyber, and pages 13.2.2 and 13.2.3, section 2.2 Database FALNET)***.

23.6. ExpressiveSystems and Yamagishi are analogous art because they are both directed to the art of logic design systems.

23.7. The motivation to use the art of Yamagishi with the art of ExpressiveSystems would have been obvious given the statement in Yamagishi that the database allows much closer integration among tools than other commercially available frameworks ***(page 13.2.3, left-side of page, lines 1 – 3)***, and the expressed benefit in ExpressiveSystems of the software to save designers time and effort by simplifying the maintenance of the design ***(page 2, second paragraph)***.

23.8. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Yamagishi with the art of ExpresiveSystems to obtain the claimed invention.

24. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference) in view of IEEEVerilog (IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language, 1995, IEEE).

24.1. Claim 20 is a dependent claim of claim 18, and thereby inherits all of the rejected limitations of claim 18.

24.2. The art of ExpressiveSystems is directed to logic design ***(page 2)***.

24.3. The art of IEEEVerilog is directed to logic design ***(page iii, section Introduction)***.

24.4. ExpressiveSystems does not specifically teach automatically indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter.

24.5. IEEEVerilog appears to teach automatically indicating design discrepancies occurring in the logic design that result from updating the value of the defined signal parameter ***(page 59, lines 1 – 25)***.

24.5.1. Regarding ***(page 59, lines 1 – 25)***; it is obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

24.6. ExpressiveSystems and IEEEVerilog are analogous art because they are both directed to the art of logic design.

24.7. The motivation to use the art of IEEEVerilog with the art of ExpressiveSystems would have been obvious since:

24.7.1. ExpressiveSystems generates Verilog computer code ***(page 24 and page 25)***, and

24.7.2. given the expressed benefit in ExpressiveSystems that the software saves designers time and effort by simplifying the maintenance of the design ***(page 2, second paragraph)***.

24.7.3. Indicating design discrepancies earlier in the design before generating Verilog computer code would save time and effort.

25. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system,

1993, IEEE 1993 Custom Integrated circuits conference) and IEEEVerilog (IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language, 1995, IEEE).

25.1. Claim 21 is a dependent claim of claim 20, and thereby inherits all of the rejected limitations of claim 20.

25.2. ExpressiveSystems appears to teach graphically indicating a bit width ***(page 24, screen displayed at top of page)***.

25.3. ExpressiveSystems does not specifically teach graphically indicating a bit width ***error***.

25.4. IEEEVerilog appears to teach indicating a bit width error ***(page 59, lines 1 – 25)***.

25.4.1. Regarding ***(page 59, lines 1 – 25)***; it is obvious that updating a signal bit width incorrectly for terminal connections would result in too few or too many bits to connect all the instances, which would cause an error.

25.5. Therefore, as discussed above, it would have been obvious to use the art of IEEEVerilog with the art of ExpressiveSystems to obtain the invention of claim 21.

26. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

26.1. Claim 22 is a dependent claim of claim 18, and thereby inherits all of the rejected limitations of claim 18.

26.2. ExpressiveSystems appears to teach that the signal parameter includes a signal bit width and the value includes a value for the signal bit width ***(page 23, on the Net Editor screen, in the field labeled New Signal(s),***

a signal is named dBus with a width of 32. This may be confirmed by examining the code generated on page 25).

27. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

27.1. Claim 23 is a dependent claim of claim 18, and thereby inherits all of the rejected limitations of claim 18.

27.2. ExpressiveSystems appears to teach that the signal parameter includes a signal bit position and the value includes a value for the signal bit position ***(page 14, section labeled 'Flexible signal handling', 'Multi-bit signals can be created with non-zero starting indexes')***.

28. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

28.1. Claim 24 is a dependent claim of claim 18, and thereby inherits all of the rejected limitations of claim 18.

28.2. ExpressiveSystems appears to teach that the signal parameter includes a bit field and the value includes a value for the bit field ***(page 14, section labeled 'Flexible signal handling', 'Multi-bit signals can be created with non-zero starting indexes', and page 8, on the signal display window, in the column labeled 'Signal Name', the signal named IO_DATA[15:0])***.

29. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over ExpressiveSystems (Internet Archive Wayback Machine, www.archive.org, search for www.expressivesystems.com, February 8, 2001) and

Yamagishi (Yamagishi, Kunihiko; Sekine, Masatoshi; A multi-representational design data capture system, 1993, IEEE 1993 Custom Integrated circuits conference).

29.1. Claim 25 is a dependent claim of claim 18, and thereby inherits all of the rejected limitations of claim 18.

29.2. Yamagishi appears to teach permitting one or more users to access the central database ***(page 13.2.1, figure 1)***.

30. Examiner's Note: Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

Conclusion

31. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

32. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russell L. Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday - Friday 9:00 AM - 5:30 PM.

33. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard can be reached on 571-272-3749. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

34. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill
Examiner
Art Unit 2123

RG

Paul L. Rodriguez 10/20/05
Primary Examiner
Art Unit 2125